CERTIFICATE OF MAILING/TRANSMISSION
I hereby certify that this correspondence is being submitted electronically via EFS Web on the date shown below.

/Steven E. Welch/ July 11, 2008

Steven E. Welch *July 11, 2008*

AMENDMENT UNDER 37 C.F.R. § 1.116
EXPEDITED PROCEDURE
EXAMINING GROUP 2154

Attorney Docket No.: 42P18331 *Patent*

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	)	
	)	
Shah et al.	)	Examiner: Larry D. Donaghue
	)	
Application No.: 10/809,077	)	Art Unit: 2154
	)	
Filed: March 24, 2004	)	Confirmation No.: 7658
	)	
For: MESSAGE CONTEXT BASED TCP TRANSMISSION	)	
	)	

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RESPONSE AFTER FINAL UNDER 37 C.F.R. §§ 1.116

Sir/Madam:

In response to the Final Office Action mailed May 23, 2008, the Examiner is respectfully solicited to consider the following remarks.

OK TO ENTER: /LD/ (07/17/2008)

CLAIM AMENDMENTS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Previously Presented) A method of transmitting packets comprising:

copying a protocol control block from a host processing system to a network protocol offload engine;

providing message information to the network protocol offload engine, the message information containing a message buffer location in a host memory;

generating one or more message contexts in the offload engine from the message information to indicate the message buffer location rather than copying the message buffer to the offload engine;

performing protocol processing at the offload engine while leaving the message buffer in the host memory; and

transmitting the message buffer in the form of at least one packet payload directly from the host memory to a network communication link, without intermediate buffering of the at least one packet payload by the offload engine, during transmission of packets by the offload engine.

2. (Original) The method of claim 1 wherein said transmitting the message buffer comprises retrieving the message buffer from the host memory via cut-through transmissions.

3. (Original) The method of claim 2 wherein said cut-through transmissions comprise direct memory access copies.

4. (Original) The method of claim 1 wherein said performing protocol processing comprises processing TCP segments.

5. (Original) The method of claim 4 wherein said performing protocol processing comprises generating TCP headers for the TCP segments.

6. (Original) The method of claim 1 further comprising freeing the one or more message contexts upon acknowledgement of the packet payload delivery.

7. (Original) The method of claim 1 further comprising providing message completion information to the host processing system to release message buffers containing the packet payload.

8. (Original) The method of claim 1 wherein said performing protocol processing comprises processing machine-readable instructions stored in a storage medium.

9. (Previously Presented) A network offload engine comprising:

a first interface to a host processor to receive a copy of a transmission control protocol block;

a second interface to a communication link to copy packet payloads from a host buffer onto the communication link; and

an engine to perform protocol processing with information from the transmission control protocol block and additional information concerning a location of the packet payloads in the host buffer, the engine to create and to send packets on the communication link according to the protocol processing with the information from the transmission control protocol block and the additional information concerning the location of the packet payloads in the host buffer while leaving the packet payloads in the host buffer, the packet payloads being directly copied from the host buffer to the communication link, without intermediate buffering of the packet payloads within the engine, to complete packet transmissions.

10. (Original) The network offload engine of claim 9 wherein the additional information concerning the location of the packet payloads in the host buffer comprises at least one message context.

11. (Original) The network offload engine of claim 9 wherein the communication

link comprises unshielded twisted pair wire for Ethernet communications.

12. (Original) The network offload engine of claim 9 wherein the direct copy of the packet payloads from the host buffer comprises a cut-through transmission of the packet payloads to the communication link of the network offload engine.

13. (Original) The network offload engine of claim 12 wherein the copy of the packet payloads from the host buffer comprises a direct memory access engine to copy the packet payloads from the host buffer.

14. (Previously Presented) A system comprising:

a host processor to manage packet transmissions from a host of the system;

a host memory to store packet payload data for the packet transmissions;

a network adapter to transmit packets from the host of the system over an Ethernet communication link; and

a network offload engine to process a transmission control protocol block that is copied from the host in combination with packet payload data addresses from the host memory for the packet transmissions, the network offload engine to directly copy the packet payload data from the host memory to the communication link during the packet transmissions without intermediate buffering of the packet payload data after copying from the host memory.

15. (Original) The system of claim 14 wherein the network adapter to transmit packets from the host of the system over an Ethernet communication link further comprises an unshielded twisted pair wire.

16. (Original) The system of claim 14 wherein the packet payload data addresses from the host memory comprise one or more message contexts.

17. (Previously Presented) An article comprising:

a storage medium comprising machine-readable instructions stored thereon to: perform protocol processing at an offload engine while leaving a packet payload of a packet in a host memory;

access one or more message contexts that contain the packet payload address from the host memory to complete the protocol processing; and

transmit the packet payload directly from the host memory to a communication link, without intermediate buffering of the packet payload within the offload engine, during transmission of the packets by the offload engine.

18. (Original) The article of claim 17 wherein the storage medium further comprises machine-readable instructions to free message contexts upon receiving an acknowledgement of payload delivery.

19. (Original) The article of claim 18 wherein the storage medium further comprises machine-readable instructions to instruct the host processing system to release message buffers of the host memory upon receiving the acknowledgement of payload delivery.

20. (Original) The article of claim 17 wherein the storage medium further comprises machine-readable instructions to instruct the host processing system to perform protocol processing for TCP segments.

21. (Original) The article of claim 20 wherein the storage medium further comprises machine-readable instructions to instruct the host processing system to generate a TCP header for the TCP segments.

REMARKS

Claims 1-21 remain pending in the instant application. All claims presently stand rejected. No claims are amended herein. Reconsideration of the pending claims is respectfully requested.

Claim Rejections – 35 U.S.C. § 103

Claims 1-5, 8-10, 12-14, 16-17 and 20-21 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Boucher et al. (US Patent Publication US 2002/156927 A1) in view of Ziai et al. (7,017,042). The rejections are respectfully traversed.

Claims 6, 7, 15, 18, and 19 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Boucher et al. in view of Ziai and purported Admitted Prior Art. The rejections are respectfully traversed.

"To establish prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. All words in a claim must be considered in judging the patentability of that claim against the prior art." M.P.E.P. § 2143.03.

Independent claim 1 recites, in pertinent parts,

> performing protocol processing at the offload engine **while leaving the message buffer in the host memory;**
> **transmitting the message buffer in the form of at least one packet payload** directly from the host memory to a network communication link, **without intermediate buffering** of the at least one packet payload **by the offload engine**, during transmission of packets by the offload engine.

Applicants respectfully submit that independent claim 1 is patentable for the two independent reasons discussed below.

Independent Reason #1

Firstly, the combination of Boucher and Ziai fails to disclose, teach, or suggest performing protocol processing at an offload engine **while leaving** the message buffer in host memory. The Office Action cites INIC/CPD 30 illustrated in FIG. 2 of Boucher as corresponding to the claimed "offload engine" (which includes processor 55 as

illustrated in FIG. 5) and cites the communication control block (CCB) as corresponding to the claimed message contexts. However, Boucher discloses,

> Guided by the CCB, the **processor 55 moves network frame-sized portions of the data from the source in host memory 35 into its own memory 60** using DMA, as depicted by arrow 99. The processor 55 **then** prepends appropriate headers and checksums to the data portions, and transmits the resulting frames to the network 25, consistent with the restrictions of the associated protocols.

Boucher, para. [0049]. This portion of Boucher fails to teach or suggest performing protocol processing within CPD 30 while leaving the frame-sized portions of the data in host memory 35. Thus, Boucher fails to teach or suggest performing protocol processing at an offload engine **while leaving** the message buffer in host memory.

Similarly, Ziai also fails to teach or suggest the very same element. In fact, Ziai discloses,

> Referring to FIGS. 3a and 3c, outbound IP packets begin as application data within the system memory 307. The **application data is then transferred (e.g. via Direct Memory Access) 301c by the system CPU/chipset 306 to the TCP/IP processors 305** (e.g., by writing the application data into the NOM 304 from the CPU/chipset 306 and reading the application data from the NOM 304 by the TCP/IP processors 305). **In alternate embodiments the application data may be written into the outbound network interface 301. Then, the TCP/IP processors 305 perform TCP/IP processing 302c on the application data**. In the outbound direction, TCP/IP processing is the addition of the TCP header 102 at the transport layer 109 and the addition of the first IP header 103 at the network layer 110, consistent with the TCP and IP protocols. **In the outbound direction, TCP/IP processing results in the creation of an IP packet**. In addition, the IP packet is checked to determine if IPSec processing is required. Thus, after TCP/IP processing, an IP packet is stored in NOM 304. In alternate embodiments the IP packet may be sent to the outbound network interface 301.

Ziai, col. 5, lines 4-23. This portion of Ziai teaches that an IP packet is not created until TCP/IP processing is performed and further that TCP/IP processing is not performed until after application data is transferred from system memory 307 into network protocol offload chip 300. Even in the disclosed "alternative embodiment" where the application data is buffered in network interface 301, Ziai still discloses that TCP/IP processing to create an IP packet is not performed until after application data is transferred into

network protocol offload chip 300. Consequently, Ziai also fails to teach or suggest performing protocol processing at an offload engine **while leaving** the message buffer in host memory, where the message buffer stores the "packet payload."

Independent Reason #2

Secondly, the combination of Boucher and Ziai fails to disclose, teach, or suggest transmitting a message buffer directly from host memory to a network communication link without intermediate buffering by an offload engine. The Office Action acknowledges that "Boucher et al. did not expressly teach without having intermediate buffering of the payload in the offload engine." *Office Action* mailed 5/23/08, page 2. Consequently, the Office Action cites Ziai as teaching this missing element. Ziai in fact discloses,

> Referring to FIGS. 3a and 3c, **outbound IP packets begin as application data within the system memory 307**. The application data is then transferred (e.g. via Direct Memory Access) 301c by the system CPU/chipset 306 to the TCP/IP processors 305 (e.g., by writing the **application data** into the NOM 304 from the CPU/chipset 306 and reading the application data from the NOM 304 by the TCP/IP processors 305). In alternate embodiments the **application data** may be written into the outbound network interface 301. **Then, the TCP/IP processors 305 perform TCP/IP processing 302c on the application data**. In the outbound direction, TCP/IP processing is the addition of the TCP header 102 at the transport layer 109 and the addition of the first IP header 103 at the network layer 110, consistent with the TCP and IP protocols. **In the outbound direction, TCP/IP processing results in the creation of an IP packet**. In addition, the IP packet is checked to determine if IPSec processing is required. Thus, after TCP/IP processing, an **IP packet** is stored in NOM 304. In alternate embodiments the **IP packet** may be sent to the outbound network interface 301.

Ziai, col. 5, lines 4-23. Accordingly, this portion of Ziai discloses that IP packets are only created after "application data" is transferred and buffered either in network offload memory 304 or network interface 301 of network protocol offload chip 300 (see FIG. 3A of Ziai). Once the application data has been transferred into TCP/IP processing is performed to create an IP packet, which is subsequently sent. Consequently, Ziai discloses a technique where application data is **intermediately buffered by network protocol offload chip 300** (i.e., either within network offload memory 304 or within network interface 301).

Consequently, the combination of Boucher and Ziai fails to teach or suggest all elements of claim 1, as required under M.P.E.P. § 2143.03. Independent claims 9, 14, and 17 each include one or both nonobvious elements as independent claim 1. Accordingly, Applicants request that the instant §103(a) rejections of claims 1, 9, 14, and 17 be withdrawn.

The dependent claims are nonobvious over the prior art of record for at least the same reasons as discussed above in connection with their respective independent claims, in addition to adding further limitations of their own. Accordingly, Applicants respectfully request that the instant § 103 rejections of the dependent claims be withdrawn.

CONCLUSION

In view of the foregoing remarks, it is believed that the applicable rejections have been overcome and all claims remaining in the application are presently in condition for allowance. Accordingly, favorable consideration and a Notice of Allowance are earnestly solicited. The Examiner is invited to telephone the undersigned representative at (206) 292-8600 if the Examiner believes that an interview might be useful for any reason.

CHARGE DEPOSIT ACCOUNT

It is not believed that extensions of time are required beyond those that may otherwise be provided for in documents accompanying this paper. However, if additional extensions of time are necessary to prevent abandonment of this application, then such extensions of time are hereby petitioned under 37 C.F.R. § 1.136(a). Any fees required therefore are hereby authorized to be charged to Deposit Account No. 02-2666. Please credit any overpayment to the same deposit account.

Respectfully submitted,
BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP

Date: July 11, 2008

/Cory G. Claassen/

Cory G. Claassen
Reg. No. 50,296
Phone: (206) 292-8600

1279 Oakmead Parkway
Sunnyvale, CA 94085-4040